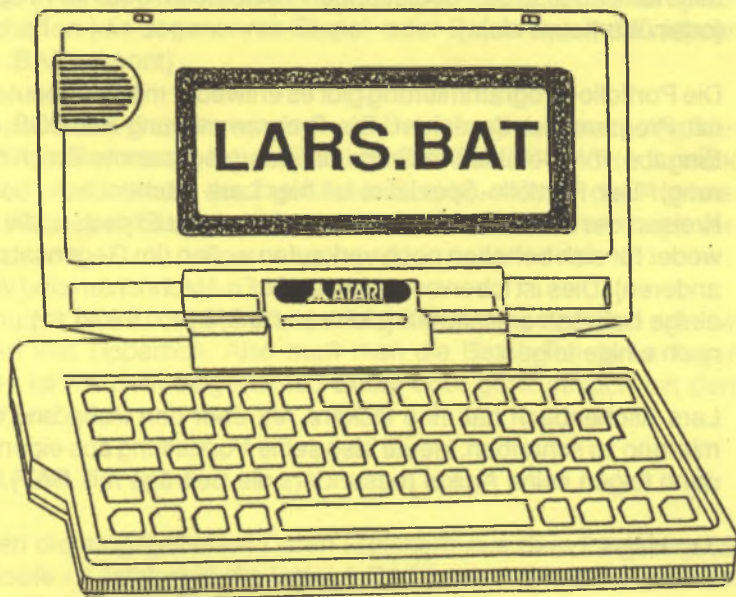


PoFo-NACHRICHTEN

November 1996

Nummer 12

Stapeldateien-Programmierung



Inhalt:

Vorwort, Impressum	2
Batch-Stapelverarbeitung	3
Sehr geehrter Leser!	4
HighBatcher L. A.	5
Die Entwicklung	6
.BAT - PoFo oder PC? (1)	8
Batchdateien in der Literatur	15
Die 126 Byte Grenze	17
.BAT - Defragmentierung	18
Programmsammlung KAP (3)	19
Anzeigen	20

Vorwort

Heute geht es um Programme und Programmierung: bei dem Portfolio gibt es grundsätzlich zwei Arten von Programmen: interne und externe. Die internen Programme sind im Handbuch ausführlich beschrieben, die externen manchmal in den begleitenden Textdateien oder im Programm selbst (oder überhaupt nicht).

Die Portfolio-Programmierung gibt es entweder mit eingebautem DOS oder mit Programmier-Sprachen. Die Programmierung mit DOS erfolgt durch Eingabe von Befehlen in Stapeldateien (sogenannte Batch-Programmierung). Der Portfolio-Spezialist ist hier Lars Aschenbach, bekannt in den Kreisen der PoFo-User als einer der wenigen Experten, die sein Wissen weder für sich behalten noch verkaufen wollen (im Gegensatz zu manchen anderen). Dies ist lobenswert. In den PoFo-Nachrichten sind von ihm schon einige Beiträge erschienen (siehe Hefte 5, 8, 9, 10 und 11) und es werden noch einige folgen.

Lars Aschenbach hat eine lockere Art, über den PoFo und die Programmierung zu schreiben. Heute also seine Vorstellung aus eigener Hand, und dann folgen seine Artikel (besonders die Beiträge mit .BAT).

Jan Hájek

Impressum

Autoren dieser Ausgabe:

Lars Aschenbach, Jan Hájek
und Mitglieder vom KAP (Klub Atari Portfolio) aus Prag

Copyright © PoFo-Nachrichten, J. Hájek, Praha 1996

Herausgeber: A A, Prag 1, Reihe PAP (Palmtop Atari Portfolio)

Satz: MQ-Mac, Druck: O. Faster, Praha 2 (Tschechische Republik)

Redaktion und Vertrieb: ABAK, Am Hedreich 5, D-44225 Dortmund

Batch - Stapelverarbeitung

Der Rechner versteht nur eingegebene Befehle, die er sofort ausführt. Da es sich oft um eine ganze Reihe von Befehlen handelt, die nacheinander eingegeben werden müssen (und meistens immer die gleichen), kann man diese Befehle in einen Stapel zusammenfassen und dadurch eine spezielle Datei schaffen (die sogenannte Stapel- oder Batch-Datei, die man an der Endung .BAT erkennt).

Die Befehle der Batch-Dateien werden vom Betriebssystem nacheinander verarbeitet. Die bekannteste Batch-Datei ist AUTOEXEC.BAT (automatic execute batch - selbständig durchgeführter Stapel), mit der der Rechner seine Arbeit nach dem Starten anfängt (soweit diese Datei überhaupt vorhanden ist).

Für den Benutzer ist es vorteilhaft, wenn er mit den Stapeldateien arbeitet - dies erspart viel Tipparbeit. Also muß man die Batch-Dateien anlegen können. Dies ist mit dem eingebauten Editor oder ganz einfach mit dem COPY-Befehl möglich:

```
COPY CON A.BAT
```

Damit werden die eingegebenen Daten (Zeichen) aus dem Eingabegerät CON (= console - Tastatur) in die Datei A.BAT kopiert. Die Datei wird am Ende mit Ctrl-Z abgeschlossen und es erscheint die Meldung:

```
1 Datei kopiert
```

Damit ist die Batch-Datei A.BAT fertig und wird durch einfaches Eingeben des Buchstaben A aufgerufen (mit Eingabe-Taste bestätigen). Die Bezeichnung A ist hier als Beispiel aufgeführt. Stattdessen kann ein beliebiger Dateiname stehen (bis zu acht Buchstaben lang, wie im MS-DOS üblich).

Die Stapel-Datei-Programmierung ist nicht Portfolio-spezifisch, aber gehört dazu (da im Portfolio das Betriebssystem MS-DOS benutzt wurde, das die Stapelverarbeitung unterstützt). Es handelt sich also um eine Programmierung mit den von MS-DOS (bzw. DIP-DOS) vorgegebenen Befehlen.

Für die Benutzer, die noch nicht die Stapel-Programmierung betrieben haben, drucken wir ab und zu einige .BAT-Dateien, meistens mit einer Erklärung, wie diese funktionieren. Es gibt auch eine höhere Schule der Stapel-Programmierung, da es nämlich etliche Tricks gibt, mit denen man kurze und effektive Stapel-Dateien schreiben kann. Auch solche komplizierteren Batch-Dateien werden in den Po-Fo-Nachrichten veröffentlicht.

Nach Unterlagen von KAP/Prag

Sehr geehrter Leser!

Ich möchte diesen Platz nutzen, um ein paar Worte über mich und meine Artikel zu verlieren:

Mein Name ist Lars Aschenbach und ich benutze den PoFo seit April'93. Seit dieser Zeit habe ich viel mit den Geräten erlebt (mittlerweile besitze ich vier Stück) und dieses in vielen Artikeln manifestiert.

Sie haben meine Texte vielleicht schon im "mobil magazin", in der "Atari inside", im Diskettenmagazin "P&P", in den "PoFo-Nachrichten" oder in dem "Pofo Info" gelesen. Diese Häufigkeit ist für Sie vielleicht bereits etwas nervig, aber nicht jeder Leser liest auch die anderen Magazine, weshalb meine Artikel von all diesen Magazinen verwendet werden!

Da ich auch nach drei Jahren immer noch aktiv bin und meine Erfahrungen niederschreibe, werden Sie auch bald neuere Artikel an dieser Stelle zu lesen bekommen!

Dennoch werde ich die Überschneidungen nicht zu unterbinden versuchen, damit eine breite Leserschaft erreicht werden kann, die meine Erfahrungen hoffentlich nachvollziehen und nutzen kann.

Für Ihr Verständnis bedankt sich:

Lars Aschenbach
Stückenstr. 55c
D-22081 Hamburg

Fax: 040 / 299 20 90
CIS: 100542.3711
Internet: 100542.3711@compuserve.com

HighBatcher L. A.

Herzlich willkommen zu dieser Serie über Batchprogrammiertechniken, die ich hiermit einleiten möchte.

Obwohl ein PC mit DOS 6.x zuhause auf mich wartet, bin ich doch mehr mit dem Atari Portfolio (PoFo) verheiratet und erstelle meine Batchdateien mit ihm. Kennern wird das wie ein Rückfall in die Steinzeit vorkommen, da das DOS des PoFo auf der Version 2.11 basiert. Gottlob ist es mit einem Schuß DOS 3.30 gewürzt, so daß es sich damit ganz gut arbeiten läßt.

Aber es sind gerade die scheinbar geringen Möglichkeiten, die mich so reizen und zu einer gewissen Programmierakrobatik treiben. Vielleicht sind meine Erkenntnisse nicht gerade die neuesten und erledigen sich durch neuere DOS-Versionen, aber sie funktionieren zumindest immer auf dem PoFo und ab DOS 3.30.

Mir liegt auch mehr daran, mit dieser Serie nachvollziehbare Beispiele zu liefern und zu Eigenentwicklungen anzuregen. Die Langlebigkeit dieser Serie liegt besonders in Euren Händen, liebe Leser, weil ich gerne auf Eure Vorschläge und Anregungen eingehe. Trotzdem kann ich nicht garantieren, für alles eine Lösung zu finden. Ich würde mich daher auch sehr freuen, wenn sich weitere Autoren anfinden, die ebenfalls ihr Wissen preisgeben und diese Serie ergänzen.

Genug der Vorstellung, schickt Eure Ideen an:

Lars Aschenbach
Stückenstr. 55c
D-22081 Hamburg

Und schreibwütige Autoren kontaktieren bitte die Redaktion.

"Gut gebatcht ist halb gepatcht!" meint Euer

Lars Aschenbach

KnowHow: Batchprogrammierung

Die Entwicklung

... von Batchdateien ist nicht immer einfach, ganz besonders dann nicht, wenn sie sehr trickreich ausfallen sollen. Durch das berühmte Debuggen, die Fehlersuche, geht gewöhnlich die meiste Zeit verloren. Dabei wird von jedem Lehrbuch empfohlen, das ECHO nicht auszuschalten und an kritischen Stellen PAUSE-Befehle einzubauen, damit die Fehlersuche vereinfacht wird.

Während der Testphase ist dagegen auch nichts einzuwenden, aber was das an Arbeit bedeutet, wenn man diese Zeilen nach den erfolgreichen Tests löschen kann, das verschweigen die Lehrmeister geflissentlich!

Da meine Routinen immer komplexer zu werden scheinen und ich nicht mehr Zeit als bisher in die Fehlersuche investieren will, suchte ich eine variable Lösung, damit ich die Batchdateien nicht nachträglich überarbeiten muß. Nach reiflicher Überlegung kam ich auf die Idee, anstelle der Befehle Variablen zu verwenden, weil diese ja beliebig undefiniert werden können!

So ersann ich die Variable O, die entweder mit dem Wort ON oder OFF definiert sein soll, damit ich zu Beginn des Batchprogramms das ECHO an- oder abschalte:

```
@ECHO %O%
```

und ich erfand die Variable &, die mit dem Befehl PAUSE oder REM definiert sein soll, damit ich an kritischen Stellen bei Bedarf eine PAUSE einlegen kann oder nicht:

```
%&%
```

Um beide Variablen gleichzeitig auf den jeweiligen Stand zu schalten: testen oder nicht testen, schuf ich dann PHASE.BAT, welche die Variablen & und O mit jedem Aufruf umschaltet:

```
@echo off
if %1.==?. if not %0%.==. goto off
for %%a in (on off) do if not %0%.==%%a. set o=%%a
for %%a in (pause rem) do if not %%a.==%%a. set &=%%a
:off
echo Die Testphase ist %0%!
```

Zukünftige Batchdateien müssen also mit @ECHO %O% beginnen und die PAUSE- Zeilen, die man zu Testzwecken einbringt, müssen durch %% gestaltet werden. Dazu mal eine Testroutine, die Ihr unter dem Namen TEST.BAT abspeichern könnt:

```
@echo %O%
echo Dies ist eine unwichtige Zeile!
%%&%
echo *Zeile ohne Wert*
```

Ruft nun abwechselnd PHASE.BAT und TEST.BAT auf, um die Wirkung von PHASE.BAT zu ergründen.

Wenn also demnächst meine Batchdateien mit @ECHO %O% beginnen sollen, so sollte die Testphase möglichst OFF-geschaltet sein.

Den Zustand der Testphase könnt Ihr auch jederzeit durch den Aufruf PHASE? erfragen. Ansonsten werdet Ihr auch nach jedem Aufruf über den aktuellen Stand informiert!

Zum Schluß noch eine Bitte:

Wer Ideen, Fragen, Probleme und Kritiken zum Thema DOS, DIP-OS und Batchdateien hat, möchte sie doch bitte hier schildern, damit wir diese umsetzen, beantworten, lösen oder verreißen (letzteres gilt nur für Kritiken!) können!

Es bedankt sich:

Lars Aschenbach

.BAT - PoFo oder PC? (1)

Laufe ich auf dem PoFo oder dem PC?

Diese Frage müssen sich Batchdateien stellen und selbst beantworten können, wenn sie auf beiden Computern laufen sollen. Denn es gibt leider ein paar Unterschiede in der Batchsyntax zwischen PoFo und PC!

So kann jede einzelne der folgenden Batchzeilen dem PC unter bestimmten Umständen Probleme bereiten:

```
IF %1==L ECHO Der Buchstabe L!  
CD..>NUL  
MD C:\DOS>NUL  
FOR %%A in (C:\DOS\*.*) DO TYPE C:\DOS\%%A  
TYPE C:\CONFIG.SYS /P  
APP /E  
OFF
```

Sie führen zu Fehlermeldungen und bringen damit Daten in Gefahr, da die Batchdatei, die nur eine dieser Zeilen beinhaltet, mit einem falschen Resultat endet. Zumindest tritt das sehr oft beim Einsatz auf dem PC ein. Für den PoFo sind diese Batchzeilen dagegen vollkommen korrekt und werden auch, beim Einsatz in Batchdateien wohlgermerkt, ordentlich ausgeführt.

Um aber diese Batchzeilen auf dem PC einsetzen zu können, müssen diese so geschrieben werden:

```
IF %1.==L. ECHO Der Buchstabe L!  
IF EXIST ..NUL CD ..  
IF NOT EXIST C:\DOS\NUL MD C:\DOS  
FOR %%A IN (C:\DOS\*.*) DO TYPE %%A  
TYPE C:\CONFIG.SYS | MORE  
EDIT - Anm.: MS-DOS als DOS vorausgesetzt  
OFF - Anm.: Dazu gibt es keinen vergleichbaren Befehl beim PC
```

In dieser Schreibweise sind aber leider nicht alle Zeilen auch für den PoFo gültig und würden bei ihm zu Fehlermeldungen führen.

Eine gemeinsame Schreibweise, die auf beiden Geräten gültig und unproblematisch ist, gibt es immerhin für die ersten drei Batchzeilen:

```
IF %1.==L. ECHO Der Buchstabe L!  
IF EXIST ..\NUL.* CD ..  
IF NOT EXIST C:\DOS\NUL.* MD C:\DOS
```

Doch bei den restlichen vier Zeilen fand ich keinen gemeinsamen Nenner und mußte daher prüfen, vor dem Einsatz dieser Zeilen, ob die Syntax vom PoFo oder vom PC anzuwenden ist.

Angeregt durch DR-DOS und Novell DOS half ich mir deshalb mit den Variablen OS (Operating System) und VER (Version) weiter, die ich dann auf dem jeweiligen Computer von Hand setzte. So nahm ich Teile der Bildschirmmeldung auf, die sich aus der Anwendung des Befehles VER ergeben, um zwischen PC und PoFo unterscheiden zu können.

```
Beim PoFo: SET OS=DIP-OS  
            SET VER=2.11
```

```
Beim PC:   SET OS=MS-DOS  
            SET VER=6.22
```

Außerdem fügte ich dann die jeweils zwei gültigen Zeilen noch in die entsprechende AUTOEXEC.BAT meines PCs und des PoFos ein.

Nun kann ich in meinen Batchdateien die Variable OS abfragen und die für das jeweilige DOS gültige Batchzeile zur Ausführung kommen lassen:

```
IF %OS%.==DIP-OS. FOR %%A IN (C:\DOS\*.*) DO TYPE C:\DOS\%%A  
IF NOT %OS%.==DIP-OS. FOR %%A IN (C:\DOS\*.*) DO TYPE %%A  
  
IF %OS%.==DIP-OS. TYPE C:\CONFIG.SYS /P  
IF NOT %OS%.==DIP-OS. TYPE C:\CONFIG.SYS | MORE
```

```
IF %OS%.==DIP-OS. APP /E  
IF NOT %OS%.==DIP-OS. EDIT
```

```
IF %OS%.==DIP-OS. OFF  
IF NOT %OS%.==DIP-OS. ECHO Bitte nun den Computer ausschalten!
```

Ich teste mit diesen Zeilen nur auf den Variableninhalt DIP-OS ab, da DR-DOS und Novell-DOS von Haus aus eine Variable OS besitzen, die mit DR-DOS bzw. NWDOS definiert ist. Es reicht ja aus zu prüfen, ob die Variable OS mit DIP-OS definiert wurde, damit die PoFo-Syntax angewendet wird!

Somit kann ich also tatsächlich Batchdateien schreiben, die sowohl auf dem PC als auch auf dem PoFo lauffähig sind!

Doch leider ist damit das Problem noch nicht ganz gelöst:

Keine Batchdatei kann sich mit absoluter Sicherheit darauf verlassen, daß die Variable OS vom Bediener angelegt wurde. Fehlt diese Variable, oder ist sie nicht mit DIP-OS definiert, so wird immer die vom PC gültige Syntax verwendet und das ohne Rücksicht auf Verluste und Fehlermeldungen!

Wie eingangs erwähnt, muß sich jede Batchdatei fragen, ob das zugrundeliegende Gerät ein PoFo oder ein PC ist. Diese Frage muß die Batchdatei dann auch noch selbst beantworten können, damit es keine Probleme gibt!

Deshalb habe ich verschiedene Techniken ausprobiert und bin auch zu sehr umständlichen, aber funktionierenden Lösungen gekommen. Weil die nun wirklich nicht akzeptabel waren, habe ich dann für ein halbes Jahr von diesem Thema Abstand genommen.

Vor einigen Tagen dann, ich beschäftigte mich gerade intensiv mit dem Verhalten der Parameter %0 und %1 auf dem PoFo, öffnete mir Kollege Zufall die Augen:

Starte ich eine Batchdatei, z.B. TEST.BAT, auf dem PoFo, indem ich ihren Namen TEST inkl. der Extension .BAT eingebe, so wird der Name im Parameter %0 und die Extension im Parameter %1 gespeichert:

PoFo: %0 = TEST %1 = .BAT

Starte ich auf dieselbe Weise beim PC, so werden der Name TEST und die Extension .BAT zusammen im Parameter %0 gespeichert. Ein Parameter %1 existiert dann nicht:

PC : %0 = TEST.BAT %1 =

Wer genau hinschaut, wird also deutliche Unterschiede zwischen den Parametern beim PoFo und beim PC feststellen!

Demnach muß es immerhin möglich sein zu prüfen, ob die Batchdatei unter DIP-OS gestartet wurde oder nicht, wenn ich die Batchdatei sich selbst neustarten lasse und den Parameter %1 überprüfe!

Und tatsächlich kann ich das mit der Batchdatei DIPOS.BAT beweisen:

```
01: @echo off
02: rem Ist die Batchdatei unter DIP-OS gestartet worden oder nicht ?
03: rem
04: rem Selbstaufruf, wegen Trennprüfung beim DIP-OS:
05: rem
06: if not %0==dlpOs.BaT if not %1==.BaT. dlpOs.BaT
07: rem
08: rem Ergibt folgende Parameter:
09: rem
10: rem bei DIP-OS: %0 = dlpOs        %1 = .BaT
11: rem bei MS-DOS: %0 = dlpOs.BaT    %1 =
12: rem
13: if %1==.BaT. echo Ich laufe unter DIP-OS !
14: if %1==.BaT. set OS=DIP-OS
15: if %1==. echo Ich laufe NICHT unter DIP-OS !
16: if %1%os%==. set OS=MS-DOS
```

Die Zeilennummern nebst Doppelpunkt sind nicht abzutippen! Die wechselnde Schreibweise ist zwar lästig, aber absolut notwendig! DIPOS.BAT sollte durch die Eingabe von "DIPOS" oder "dipos" gestartet werden!

Weil bei diesem Start noch die Extension .BaT fehlt, was ja vollauf beabsichtigt ist, fallen die Prüfungen in Zeile 06: negativ aus und sorgen für den Neustart inklusive der Extension: dlpOs.BaT

Dieser Neustart führt zu unterschiedlichen Reaktionen bei PC und PoFo, weshalb wir diese einmal getrennt betrachten:

Beim PC wird ja der Parameter %0, hier: dlpOs.BaT, unbehellig gelassen, weshalb die erste IF NOT-Abfrage in Zeile 06: greift und somit kein weiterer Neustart mehr notwendig ist. Weil kein Parameter %1 existiert, der .BaT lauten könnte, kommt die Zeile 15: zur vollen Geltung und gibt kund, daß DIPOS.BAT nicht unter DIP-OS (PoFo) gestartet wurde!

Die Zeile 16: hält dann dieses Ergebnis nach Möglichkeit in der Variablen OS (für Operating System) fest, sofern bislang keine Variable OS existiert. Damit können nachfolgende Batchdateien das OS prüfen, ohne erst einen Neustart veranlassen zu müssen. Existierte aber bereits eine Variable OS, bei DR-DOS und Novell-DOS sollte das ja der Fall sein, so bleibt die bisherige Definition von OS erhalten.

Beim Neustart auf dem PoFo findet dagegen die Aufteilung von dlpOs.BaT in dlpOs und .BaT statt. Somit lautet der Parameter %0 dlpOs und der Parameter %1 .BaT. Diese Aufteilung und Umwandlung findet übrigens noch vor der Ausführung der Zeile 01: statt! Nun greift die zweite IF NOT-Abfrage der Zeile 06: und verhindert einen weiteren Neustart. Da jetzt der Parameter %1 existiert und zudem noch .BaT lautet, sind die Bedingungen der Zeile 13: erfüllt und es wird gemeldet, daß DIPOS.BAT unter DIP-OS (PoFo) gestartet wurde. Hier hält dann die Zeile 14: das Ergebnis für nachfolgende Batchdateien in der Variablen OS fest. Voila!

Nun kann man entweder DIPOS.BAT vor den eigentlichen Batchdateien starten, damit die Variable OS auch existiert, oder man setzt seine Routine in das folgende Grundgerüst ein:

```
01: @echo off
02: for %%a in (%bdv%. %bdy%.) do if %%a==. goto err
03: if not %0==%BDV%%BDY%bAtCh.BaT if not %1==.BaT.
%BDV%%BDY%bAtCh.BaT %1 %2 %3 %4 %5 %6 %7 %8 %9
```

```
04: if %1.==.BaT. set OS=DIP-OS
05: if %1%os%.==. set OS=MS-DOS
06: if %1.==.BaT. shift
07: rem
08: rem Hier muß dann die eigentliche Routine eingebunden werden!
09: rem
97: :err
98: for %%a in (%bdv%. %bdy%.) do if %%a==. echo Eine der Variablen
BDV und
BDY existiert nicht!
99: :off
```

In der Zeile 03: muß dann anstelle von bAtCh.BaT und bAtCh.BaT der endgültige Name der Batchdatei eingetragen werden. Bitte achtet dabei auf die wechselnde Schreibweise, da sie sehr, sehr wichtig ist!

Neu sind dabei die Variablen BDV und BDY:

Hinter diesen Variablen, die in vielen meiner Batchdateien vorkommen, verbergen sich das BatchDrive (das Laufwerk AUF DEM sich meine Batchdateien befinden) und das BatchDirectory (das Verzeichnis IN DEM meine Batchdateien stehen).

So habe ich BDV durch die AUTOEXEC.BAT mit SET BDV=C: und BDY mit SET BDY=\BATCH\definiert. Damit das DOS meine Batchdateien auch jederzeit wiederfindet, habe ich dann noch das Verzeichnis C:\BATCH mit in die PATH-Anweisung aufgenommen: PATH=C:\DOS;C:\BATCH

Das ganze hat folgende Vorteile:

Will ich meine Batchdateien veröffentlichen, in ein anderes Laufwerk und Verzeichnis verlagern oder auf einen anderen PC übertragen, so müssen weder ich noch ein anderer User die Laufwerks- und Verzeichnisangaben in den ganzen Batchdateien ändern, damit sie unter den neuen Bedingungen auch funktionieren. Es muß lediglich dafür Sorge getragen werden, daß die Variablen BDV, BDY und PATH angepaßt oder bereitgestellt werden:

Dadurch haben alle User deutlich weniger Tipparbeit und sind flexibel!

Wer dieses Grundgerüst in Zukunft für seine Batchdateien verwendet, kann nun auf zweierlei Wegen abprüfen, ob die Batchdatei auf dem PoFo läuft oder nicht:

Entweder man prüft die Variable OS also auch weiterhin auf den Inhalt DIP-OS ab, oder man prüft den Parameter %0, der dann beim PoFo .BaT heißen muß, wenn die eigentliche Routine zum Zuge kommt. Weil die Variable OS nicht in allen Fällen existiert, oder angelegt werden kann, ist die Prüfung des Parameters %0 auf .BaT besonders zu empfehlen!

Warum die Variable OS überhaupt von der Firma Digital Research eingeführt wurde und ich mir diese zunutze mache:

So manches Programm will nur partout unter MS-DOS oder PC-DOS funktionieren und fragt daher die Betriebssystemversion ab. Lautet diese z.B. nicht "MS-DOS 4.0", so stellt das Programm seine Arbeit unverzüglich ein.

Den Entwicklern von DR-DOS, welches später zu NovellDOS wurde, war dies natürlich ein Dorn im Auge, weshalb sie den Käufern eine einfache Manipulationsmöglichkeit dieser Betriebssystemversion boten:

Mittels SET-Befehl kann man dort die Variablen OS und VER (für Version) auf die gewünschten Werte einstellen, damit das gewünschte Programm überlistet wird und läuft.

Weil das Rad nicht zweimal erfunden werden muß, englischsprachige Variablen bei der internationalen Verbreitung meines Wissens hilfreich sind und ich eine straffe Organisation bevorzuge, verwende also auch ich die Variable OS!

Die Variable VER spielt zwar derzeit noch keine Rolle, aber ich werde sie dann nutzen, wenn ich auf dem PC die externen Programme des jeweiligen DOS verwenden möchte. Auch für den PoFo ist diese Variable VER noch unbedeutend und muß nicht zwingend angelegt werden. Aber wer weiß?

Lars Aschenbach

Batchdateien in der Literatur

In Zeiten zunehmender Verfensterung ist es natürlich nicht leicht, Bücher zum Thema Batchdateien im Handel zu finden. Aber mit etwas Glück und Geduld kann man gesuchte Bücher von Privat erwerben.

Leider gibt es nur recht wenige Bücher in deutscher Sprache dazu und diese sind meist nur für den wißbegierigen Beginner interessant:

“So geht’s! Utilities und Batchdateien” von Rudi Kost, Josef Steiner und Robert Valentin. Erschienen 1993 unter der ISBN 3-87791-467-5 im Markt & Technik Verlag.

“Batch-Dateien - Einsteigerseminar” von Dr. Ekkehard Kaier. Erschienen 1992 unter der ISBN 3-89360-618-1 im bhv Verlag.

“Batchdateien für MS-DOS 6.0 entwickeln” von Hellermann und Lemme. Erschienen 1993 unter der ISBN 3-7723-5023-2 im Franzis-Verlag.

“Stapeljobs unter MS-DOS” von Paulissen & Paulissen. Erschienen 1990 unter der ISBN 3-88963-241-6 im Hofacker Verlag.

Aus der Reihe “PC-Grundkurs” des Verlag Harri Deutsch stammt Wolfgang Webers Buch “@ DOS Batch-Programmierung”. Dieses Werk mit der ISBN 3-8171-1266-1 stammt zwar bereits von 1992, ist aber derzeit noch über den Buchhandel bestellbar!

Im Frühjahr ‘96 soll angeblich noch ein Buch mit dem Titel “Batch mal wieder!” erscheinen. Einer der Autoren soll Werner “Tikki” Küstenmacher sein. Dem Titel nach könnte es ein Buch für fortgeschrittene Batchprogrammierer werden. Hoffen wir also das beste!

Weitere Bücher, die meist ein Kapitel zum Thema Batchdateien haben, sind:

“MS-DOS 6 - Tips & Tricks” von Helmut und Manfred Tornsdorf. Erschienen 1993 unter der ISBN 3-8158-1014-0 bei Data Becker.

"MS-DOS Trainer" von Bach und Backer. Erschienen 1995 unter der ISBN 3-8043-3019-3 im Augustus Verlag.

"Das MS-DOS Kompendium" von Chris DeVoney. Erschienen 1985 unter der ISBN 3-89090-106-9 im Markt & Technik Verlag.

Aus der Rolle fallen die folgenden Bücher:

"Data Beckers große MS-DOS Toolbox" von Helmut und Manfred Torns-dorf. Erschienen 1990 unter der ISBN 3-89011-824-0 bei Data Becker. Dieses Buch bietet eine ausführliche Einführung in die Batchprogrammierung und zusätzlich viele Hilfsprogramme auf beiliegender Diskette.

"Softwarelexikon MS-DOS" von Thomas Tai und Peter Freese. Erschienen 1988 unter der ISBN 3-499-18152-5 im rororo Verlag. Es ist mehr für den engagierten MS-DOS-Anwender gedacht, da es u.a. beschreibt, ab wann welcher Schalter, welche Option und welcher Befehl in MS-DOS eingeführt wurde. Aber auch Begriffe wie Akustikkoppler usw. werden dort erklärt.

An Zeitschriften kann ich empfehlen:

1. Die Sonderhefte des PC-Weit Magazins
2. Die ersten zwei Ausgaben des Magazins "100 Tips & Tools für Windows & MS-DOS" vom Verlag VTP Fürst
3. "Was ist denn DOS?" vom KnowWare Verlag

Ein bis zwei Seiten mit DOS-Tricks und Batchdateien findet man auch monatlich in den Zeitschriften PC-Welt, PC Praxis, PC go!, CHIP, DOS und der Highscreen Highlights.

Ansonsten kann ich den Datendienst CompuServe wärmstens empfehlen, speziell das MS-DOS-Forum (GO MSDOS), da man dort die besten Batchprogrammierer antrifft. Selbst wenn eine Frage schon dreißigmal gestellt wurde, wird sie auch beim 31. Mal beantwortet! Über den PC-File-Finder von CompuServe (GO PCFF) kann man auch eine ganze Reihe von Batchdateien finden, wenn man als Schlüsselworte (Keywords) BATCH und DOS oder als Extension BAT einträgt.

Für weitere Quellen und Hinweise ist dankbar: Lars Aschenbach

Know How: Batchprogrammierung

Die 126 Byte Grenze

Selbst bei mir kommt es selten vor, daß eine Batchzeile über 126 Buchstaben lang ist. Aber neulich kam es dann doch dazu und führte zu einer interessanten Reaktion von Seiten des PoFo!

So nahm der PoFo bei Ablauf der Batchdatei zunächst nur die ersten 126 Byte als eine Zeile wahr und führte diese aus. Danach nahm er sich den Rest der Zeile zur Brust und versuchte, sie auszuführen.

Da aber der besagte Zeilenrest nicht aus Befehlen, sondern einem Text bestand, der per ECHO auf den Schirm gebracht werden sollte, quittierte der PoFo den Rest der Zeile mit der Fehlermeldung "Befehl oder Programm nicht gefunden".

Ihr könnt das z.B. mit dieser Batchdatei TEST126.BAT nachprüfen:

01: @echo off

02: echo Wer sagt, daß der PoFo kein vollwertiger PC ist, der hat erstens keine Ahnung und zweitens noch nie ernsthaft mit ihm gearbeitet! Lars Aschenbach

Die Zeilennummern nebst Doppelpunkt sind nicht abzutippen!

Wenn Ihr dann TEST126.BAT startet, so wird der Text nur bis zum Wort "gearbeitet" ausgegeben, wobei von gearbeitet nur die ersten drei Buchstaben dargestellt werden. Danach taucht dann die besagte Fehlermeldung auf!

Und was lernen wir daraus?

Eine Batchzeile darf maximal 126 Zeichen/Zahlen/Buchstaben enthalten, weil der Rest nur separat behandelt wird!

Lars Aschenbach

.BAT - Defragmentierung (Aufräumarbeiten)

Umso öfter man seine Meisterwerke überarbeitet und ergänzt, umso mehr verteilen sie sich zerstückelt auf den Datenträgern. DOS hat damit zwar keine Probleme, weil es diese Fragmente zusammensucht, braucht aber mehr Zeit dafür. Die Zeit ist ein wichtiger Faktor, wenn man wie ich mit einem 4,77 MHz (*) getakteten PoFo arbeitet.

Aus diesem Grunde ist "DEFrag.BAT" entstanden, welches der Zerstückelung Einhalt gebietet:

```
rem Verzeichnis defragmentieren
if %1.==. goto err
echo Bitte etwas Geduld ...
md \defrag>nul
copy %1\*. * \defrag>nul
for %%a in (%1\*.*) do del %1\%%a
rd %1>nul
md %1>nul
copy \defrag\*. * %1>nul
for %%a in (\defrag\*.*) do del \defrag\%%a
rd \defrag>nul
echo Defragmentierung abgeschlossen!
goto off
:err
echo Aufruf mit DEFrag \Verzeichnis!
:off
```

(*) Anm. d. Redaktion:

Hier irrt auch der Spezialist
- oder er ließ sich von den
Atari Prospekten verführen
- die richtige Taktfrequenz
ist **4,9152 MHz**: siehe dazu
den Beitrag in den PoFo-
Nachrichten Nr. 8, Seite 7
bis 11).

Diese Routine legt das Verzeichnis DEFrag an und kopiert zunächst alle Dateien des gewünschten Verzeichnisses dort hinein. Dadurch finden die Dateien wieder zueinander und werden alsbald in ihr ursprüngliches Verzeichnis rückverlegt. Vorher wurde natürlich das Verzeichnis komplett geleert, gelöscht und neu kreiert. Sobald das erledigt ist, wird das nun nicht mehr benötigte Verzeichnis DEFrag getilgt und der gesamte Vorgang für abgeschlossen erklärt.

Lars Aschenbach

Programmsammlung KAP (3)

Inhalt der dritten Diskette (KAP 3) sind graphische Programme:

SHOWPIC	MICKEY.APF	1920	21.04.93	9:20
	SHOWPIC.EXE	3578	23.05.93	23:38
PGC	PGSHOW.EXE	1788	21.04.93	9:22
	PGSHOW.DOC	4929	21.04.93	9:22
	PGCOMP.EXE	7094	7.01.91	18:36
	PGCOMP.DOC	1791	7.01.91	18:39
	PGEDIT.EXE	38141	21.04.93	9:22
	COURB.FON	19088	21.04.93	9:22
	HELVB.FON	50880	21.04.93	9:22
	PGEDIT.DOC	9437	21.04.93	9:22
	PGCSPEC.TXT	6175	5.01.91	10:37
	ADEMO.BAT	117	21.04.93	9:22
	DEMO.BAT	53	23.05.93	14:41
	BOMB.PGC	348	21.04.93	9:22
	FUJI.PGC	1385	21.04.93	9:22
	GORBY.PGC	620	21.04.93	9:22
	PORTF.PGC	1296	21.04.93	9:22
	SPIRAL.PGC	473	21.04.93	9:22
	USA.PGC	553	21.04.93	9:22
	F_DESKTO.PGC	1024	21.04.93	9:22
PGX	MKPGX.EXE	2252	21.04.93	9:22
	MKPGX.DOC	3352	21.04.93	9:22
	PGFLIX.COM	1919	21.04.93	9:22
	PGFLIX.DOC	1056	21.04.93	9:22
	MAKERT.EXE	6822	21.04.93	9:22
	PGFLIXRT.DOC	1989	21.04.93	9:22
	PGFLIXRT.HDR	1248	21.04.93	9:22
	PGXSPEC.TXT	6708	21.04.93	9:22
	DOMINO.PGX	28182	21.04.93	9:22
	HORSE.PGX	74902	21.04.93	9:22

SKETCH	SKETCH.COM	4163	21.04.93	9:25
	SKETCH.DOC	2719	21.04.93	9:25
	ANSI.SYS	1862	0.00.28	0:00
FIX0D	FIX0D.COM	253	21.04.93	9:21
	FIX0D.DOC	1086	21.04.93	9:21
TEST0D	TEST0D.COM	248	21.04.93	9:21
	TEST0D.DOC	1234	21.04.93	9:21
PGLIB	PGLIB.TXT	430	24.05.93	17:50
	PGLIB.EXE	16597	24.05.93	17:47
SNATCH	SNATCH.DOC	3590	1.01.91	23:33
	SNATCH.EXE	9958	24.05.93	17:44
	LAURA.PGF	1920	1.01.91	22:35
	WALLY.PGF	1920	1.01.91	19:24
	WW.PGF	1920	1.01.91	11:59

Weitere Verzeichnisse folgen

PortWalk - Magnetbandinterface **für 89 DM**
(siehe Berichte in PoFo-Nachrichten Nr. 2, S. 3-9)

PortEx - Portfolio-Experimentierplatine **für 49 DM**
(siehe Bericht in PoFo-Nachrichten Nr. 6, S. 3)

PortErK - Portfolio-Ersatz-Kabel **für 15 DM**
(siehe Bericht in PoFo-Nachrichten Nr. 7, S. 3-5)

Programmdisketten **KAP 1 bis KAP 10 je 5 DM**
Disketten-Format 5,25" (360 KB)

(siehe Übersicht in PoFo-Nachrichten Nr. 4, Seite 19 und genaue Verzeichnisse der Dateien in jedem Heft ab Nr. 10)

Preise verstehen sich zuzüglich Versandkosten

ABAK, Am Hedreisch 5, D-44225 Dortmund